

Versiebeheer

In AORTA 8 wordt er uitgewisseld op basis van bouwstenen. Binnen AORTA worden diverse bouwstenen gebruikt. Deze bouwstenen kunnen binnen diverse zorgtoepassingen gebruikt worden. Het kan echter zo zijn dat er binnen een bepaalde zorgtoepassing een wijziging in een bouwsteen gewenst is, maar die wijziging is niet (erg) relevant voor andere zorgtoepassingen. Een wijziging in een bouwsteen, betekent dat er een nieuwe versie van de bouwsteen ontstaat.

Om ervoor te zorgen dat er nieuwe versies van bouwstenen geïntroduceerd kunnen worden zonder dat dit impact heeft op andere zorgtoepassingen zijn er binnen AORTA regels omtrent versiebeheer.

Het mechanisme wordt in de paragraaf hieronder toegelicht. In de opvolgende paragrafen worden er voorbeelden gegeven om het mechanisme toe te lichten.

Versiebeheer binnen AORTA is gebaseerd op een stabiele minimum versie(baseline) en een mogelijk van meerdere opvolgende maar niet vereiste opvolgende versies.

Door ervoor te zorgen dat er binnen AORTA voor ieder systeem altijd een minimum versie is waarmee informatie uitgewisseld kan worden, wordt ervoor gezorgd dat er een manier is waarop ieder systeem binnen AORTA informatie kan uitwisselen. Daarnaast bestaat er een mogelijkheid om nieuwe en opvolgende versies te definiëren. In deze nieuwe versies kunnen zaken zijn toegevoegd die gewenst zijn binnen een bepaalde toepassing. Het is echter zo dat niet alle zaken zomaar ketenbreed uitgerold kunnen worden. Vandaar dat er een mogelijkheid is toegevoegd zodat zorgaanbieders informatie kunnen uitwisselen zonder dat daarvoor iedere partij binnen AORTA dit moet kunnen ondersteunen.

Het versiebeheer wordt zo toegepast, dat versies die in productie zijn niet te veel en te vaak veranderen om te voorkomen dat leveranciers van bronsystemen vaak aanpassingen moeten maken om nieuwe versies van bouwstenen te ondersteunen.

Binnen tests of POC's is het juist wel gewenst om wijzigingen in bouwstenen te kunnen testen en wil je juist flexibel zijn met wijzigingen.

Om een mechanisme te creëren dat juist stabiel is in productie en flexibel in andere omgevingen, zijn er naast technische eisen ook procedurele eisen nodig en is er een mechanisme nodig dat meerdere versies van een bouwsteen ondersteunt. Om dit te kunnen faciliteren stelt VZVZ voor om gebruik te maken van baselines.

Baselines zijn versies die in de loop der tijd ontstaan en waarvan we gezamenlijk binnen de infrastructuur afspreken dat deze erg belangrijk zijn en dat die specifieke versie van een bouwsteen-type altijd opgeleverd moet kunnen worden. Het gaat hier om een in de gehele keten geïmplementeerde versie van een bouwsteen/type. Naast de baseline kan een leverancier meerdere andere versies ondersteunen afhankelijk van de programma's en POC's waarin wordt deelgenomen.

De functie van een baseline is om één versie van een bouwsteen/type te definiëren dat ieder systeem minimaal ondersteund. Op deze manier kan je er altijd op vertrouwen dat er met een systeem gecommuniceerd kan worden.

Een baseline is een versie van een bouwsteen die volgens een werkplan afspraak altijd minimaal ondersteund moet worden. Dit is dus een minimale versie die op basis van afspraken wordt geëist.

Een baseline is echter een versie waarvan we afspreken dat iedereen deze moet kunnen uitwisselen. Het is praktisch als de versies die daarna worden gepubliceerd backwards compatible zijn met deze baseline versie, want dan kan ook de nieuwe versie worden verstuurd terwijl deze ook nog door oude systemen kunnen worden verwerkt. Dit is echter geen eis bij het vaststellen van de baseline.

Systemen zijn daarnaast vrij om bouwsteen versies te implementeren die (ver) vooruitlopen op de baseline versie van de bouwsteen-types. Zij kunnen deze nieuwe versies ook uitwisselen met systemen die deze nieuwe versies al ondersteunen. Wil er echter een systeem zonder de ondersteuning van de laatste versie van

een bouwsteen communiceren met systemen die al een veel nieuwere versie ondersteund? Dan kan er altijd worden teruggevallen op de baseline versie.

Vanwege de ontwikkelingen die in de versies van de bouwsteen-types uit de verschillende programma's plaats vinden, kan het wenselijk zijn om een nieuwe baseline vast te stellen. Er is dan een nieuwe (minimale) invulling van een bouwsteentype gewenst in de hele keten. Om deze overgang te kunnen faciliteren, zal er een situatie ontstaan waarin de bestaande baseline en de beoogde nieuwe baseline ondersteund dient te worden door de bronsystemen. Ieder bronsysteem kan in eigen tempo, maar voor een bepaalde deadline, de nieuwe beoogde baseline implementeren en uitrollen. In deze situatie dienen er twee versies ondersteund te worden. Op het moment dat alle bronsystemen zijn uitgerold met ondersteuning voor de nieuwe beoogde baseline, zal deze beoogde baseline de nieuwe baseline worden en kan de eerdere baseline in de gehele keten worden uitgefaseerd.

Om het versiebeheermechanisme toe te lichten zijn hieronder een aantal voorbeeld situaties geschetst.

Voorbeeld 1: 2 Systemen met een hoge mutatiegraad

In dit voorbeeld zijn er 2 systemen met een hoge mutatiegraad, systeem H1 en H2, die allebei snel ondersteuning van de nieuwe publicaties inbouwen. De baseline is in dit voorbeeld in het werkplan vastgesteld op versie 1.0 van de medicatieafpraak. Er bestaat nog geen nieuwe beoogde baseline van de medicatie afspraak bouwsteen.

In de nieuwste Medicatieproces publicatie wordt echter versie 2.4 al gebruikt.

Wanneer systeem H1 en H2 beide ondersteuning voor versie 2.4 inbouwen, dan kunnen zij onderling al versie 2.4 uitwisselen. Versie 2.4 is backwards compatibel met 2.1, 2.2 en 2.3 dus alle systemen die versie 2.x ondersteunen kunnen een 2.x uitwisselen.

Daarnaast dienen systeem H1 en H2 versie 1.0 van de medicatieafpraak nog te ondersteunen, omdat dit nog de bestaande baseline is.

In dit voorbeeld implementeren systemen H1 en H2 beide versie 1.0 en 2.4.

Voorbeeld 2: Systeem met een lage mutatiegraad

In dit voorbeeld is er een extra systeem tov voorbeeld 1, systeem L1. L1 is een systeem dat nog niet de laatste versie(2.4) heeft ingebouwd voor Medicatieproces vanwege andere prioriteiten en wordt geclassificeerd als een systeem met een lage mutatiegraad. Volgens de afspraken binnen het werkplan moeten alle systemen die medicatieafspraken uitwisselen wel de baseline ondersteunen. De baseline versie is 1.0. Daarom zal systeem L1 minimaal versie 1.0 geïmplementeerd moeten hebben.

Voorbeeld 3: Systeem met een matige mutatiegraad

In dit voorbeeld is er nog een systeem, systeem M1. Systeem M1 is sneller met ontwikkelen dan systeem L1, maar niet zo snel als systeem H1 en H2. Systeem M1 heeft de vereiste baseline versie van medicatieafspraken ingebouwd, versie 1.0. Daarnaast ondersteund dit systeem ook al versie 2.1. Dit is niet de laatste versie van de medicatieafpraak, maar een redelijk recente versie. Het systeem wordt daarom geclassificeerd als een systeem met een matige mutatiegraad.

Voorbeeld 4 Verandering van baseline

In voorbeeld 4 treedt er een verandering op. Er wordt nu namelijk bepaald in het werkplan dat er over moet worden gegaan naar een nieuwe baseline voor medicatieafspraken. Het plan is om versie 2.1 de baseline van medicatieafspraken te maken.

Er wordt afgesproken dat alle systemen die medicatieafspraken uitwisselen vanaf 1 oktober 2019 ook versie 2.1 moeten kunnen uitwisselen. Als alle systemen vanaf 1 oktober 2019 de beoogde baseline ondersteunen, kan dit de nieuwe baseline worden en kan er worden besloten om versie 1.0 als baseline te verwijderen. Alle partijen kunnen nu nog succesvol met elkaar communiceren, iedereen heeft namelijk versie 2.1 ingebouwd. Versie 1.0 wordt niet meer ondersteund en uitwisseling ervan wordt op het LSP gedeactiveerd. Vervolgens kan er in het werkplan een nieuwe afspraak worden gemaakt voor een nieuwe baseline.

Voor de verandering van baseline van versie 1.0 naar 2.1, zal alleen het systeem met een lage mutatiegraad, L1 uit het voorbeeld hierboven, extra moeite in de ontwikkeling van de medicatieafpraak bouwsteen moeten steken. De systemen met een matige en hoge mutatiegraad (H1, H2 en M1) moeten door de ophoging van de baseline geen extra energie in het ondersteunen van de nieuwe baseline besteden aangezien zij al versie 2.1 ondersteunen.

Voorbeeld 5 Doorontwikkeling bij systemen met een hoge mutatiegraad

In de tijd tussen nu en 1 oktober 2019 zou er ook een nieuwe medicatieproces publicatie kunnen komen waarin een nieuwe versie van medicatieafspraken, versie 3.0, wordt gepubliceerd. De systemen H1 en H2 met een hoge mutatiegraad zouden deze versie in de tussentijd ook willen gaan ondersteunen. Als zij dit ondersteunen, dan kunnen ze deze onderling uitwisselen zoals geschetst in voorbeeld 1.

Om ook nog met systemen met een matige en lage mutatiegraad te kunnen uitwisselen dienen de systemen H1 en H2 nog wel de op dat moment bestaande baseline en mogelijk beoogde baseline te ondersteunen.

Voorbeeld 6: Gebruik maken van de bouwsteen binnen een POC

Binnen een POC van eSpoed wordt een nieuwe versie van de medicatieafpraak getest. Dit wordt versie 3.1 van de medicatieafpraak.

Versie 3.1 kan geen baseline worden zolang deze alleen wordt gebruikt binnen de POC. Versie 3.1 wordt nu alleen ingebouwd door de leveranciers die meedoen met de POC. Het invoeren van deze versie heeft geen impact op wat er in productie wordt uitgewisseld.

Eisen aan systemen

Om het bovenstaande versiebeheermechanisme succes te laten werken binnen AORTA zijn er eisen gedefinieerd waaraan GBX'en zich dienen te houden. Deze eisen zijn uitgewerkt in [PvE Infra Systeemrollen].